International Journal on Applications in Engineering and Technology
Volume 8 : Issue 2 : May 2022, pp 68 – 75 www.aetsjournal.com
ISSN (Online) : 2455 - 0523

TRAFFIC SIGN DETECTION USING YOLOv3

VASANTH KUMAR C[1], UMA SHANKAR S V[2], PRAKASH P[3]

[1,2] Undergraduate student, Department of Computer Science and Engineering, Paavai College of Engineering
[3]Assistant Professor, Department of Computer Science and Engineering, Paavai College of Engineering

Abstract: *-* – The detection and recognition of traffic signs is a relatively new technological advance in intelligent vehicles. Even though this technology has yet to be implemented in our country, the need for autonomous traffic guidance in the streets is self-evident. Traffic-sign detection and recognition algorithms in their early stages face challenges such as missing traffic signs, changing traffic light states, low-resolution cameras on vehicles, difficult-to-detect road-side signs, and poor real-time performance of deep learning-based methodologies for traffic-sign recognition. In this paper, we offer the YOLOv3 traffic-sign detection and identification technique, which is based on OpenCV and written in Python. We wanted to solve issues like how standard traffic sign detection is readily influenced.

Key Words: Traffic Sign detection, YOLOv3, Patch wise Training, Deep Learning

I. INTRODUCTION

When the transportation system on the highways became occupied by engine-driven cars around the world, the demand for traffic control and navigation systems arose. To satisfy this demand, multiple traffic rules were enacted, and several types of traffic signs were developed for use on the roadsides, with the goal of guiding drivers and instilling good driving habits in order to reduce the amount of road accidents, traffic jams, and other problems. As the traffic navigation process has progressed, humans are now aided by a variety of intelligent vehicle technologies, which are being used by a few first-world countries. There is an automated navigation system that detects and recognises traffic signs, as well as voice command processing technologies. In a developing country like Bangladesh, the number of vehicles on the road is fast expanding, yet the traffic control system is still outdated. As of October 2020, the Bangladesh Road Transport Authority (BRTA) reports that the country has over 4.4 million registered vehicles. The majority of local transport drivers and assistants are illiterate or asleep to the point that they are unable or unwilling to heed traffic signs. As a result, traffic rule infractions, severe road accidents, and intolerable traffic bottlenecks have grown widespread in major cities such as Dhaka and Chittagong. It has evolved into one of the most pressing issues we face on a daily basis. As a result, the demand for an autonomous navigation aid on vehicles that can detect and recognise traffic signs along roadsides and assist drivers in safely navigating is much larger than in most other countries. Traffic-sign detection technology is primarily based on information about traffic signs, such as their shape, texture, and colour, and accurately extracts traffic sign candidates from the actual road view area. We notice many types of traffic signs all over our roadside, which can be classified into four categories: I Caution ii) Prohibition iii) Obligation iv) Information One of the most difficult issues in this discipline is obtaining a useful dataset including real-world photos of many types of traffic signs under various situations. We notice many types of traffic signs all over our roadside, which can be classified into four categories: I Warning, ii) Prohibition, iii) Obligation, and iv) Informative. One of the most difficult issues in this discipline is obtaining a useful dataset including real-world photos of many types of traffic signs under various situations. Most previous research works to detect and recognise these traffic-signs are ineffective in real-time situations because traffic-sign images were captured in ideal conditions in the majority of cases, making it difficult to develop a better system for non-ideal conditions such as the presence of different lighting conditions, environmental diversities, viewing angles, transparency, and so on. Our main goal in this research is to show a real-time detection and recognition system for traffic signs in Bangladesh. Our detection technique is reliable since we evaluated it with a dataset that includes photographs that we took under various lighting and environmental circumstances. Because we used one of the most recent object identification algorithms, YOLOv3, which has proven to be a good competitor to Fast R-CNNs and SSDs in terms of detection and performance, our detection method is real-time. We employed a camera to capture real-time traffic sign photos, which were subsequently analysed by a YOLOv3-operated software that used OpenCV in Python to process the images. We structured our working technique into four parts while working with our proposed system. To begin, we gathered 78 different photographs of traffic signs from around our neighbourhood, divided into six categories. The photos were then processed to create a dataset of 800*600 pixels in jpg format. After that, we

International Journal on Applications in Engineering and Technology
Volume 8 : Issue 2 : May 2022, pp 68 – 75 www.aetsjournal.com ISSN (Online) : 2455 - 0523

used a labelling tool to identify this image, which assisted us in creating a bounding box of images. Finally, we trained our model with our supplied dataset and put it to the test with a set of test photos. Because we used one of the most recent object identification algorithms, YOLOv3, which has proven to be a good competitor to Fast R-CNNs and SSDs in terms of detection and performance, our detection method is real-time. We employed a camera to capture real-time traffic sign photos, which were subsequently analysed by a YOLOv3-operated software that used OpenCV in Python to process the images. We structured our working technique into four parts while working with our proposed system. To begin, we gathered 78 different photographs of traffic signs from around our neighbourhood, divided into six categories. As a supplementary experiment for this study, we employed our webcam for real-time detection and recognition.

1.1 PROBLEM DEFINITION

The first planned task is part of the RoboCup Portuguese Open's autonomous driving competition. This competition models some of the issues that arise when working on autonomous driving on a small and controlled basis. It comprises of a two-lane track with two curves built up so that the cars can drive continuously around the course. Vertical traffic signs, traffic signals, two parking spaces, and traffic cones for temporary lanes and obstacles are all present. The "Vertical traffic signs detection challenge" is the task under consideration for this project.

The second proposed task is identical to the first in that it involves detecting and recognising traffic signs and lights, with the main difference being the setting. It is tested on a genuine car that is being driven on public roads. This system must be able to identify a wider range of traffic signs at a greater distance away from the vehicle, as well as diverse weather and light circumstances.

1.2 YOLOV3

Darknet-53 was created by YOLOV3 using 5 residual pieces and the concept of a residual neural network as a guide. To anticipate category outcomes, YOLOV3 uses up-sampling and fusion methods, as well as three different fusion scales to detect the target. The identification of objects of various sizes and obstructed objects has been improved, and a jump layer connection has been included to improve the convergence effect.

In the left side of the blue red numbers in the first line of each module, the number of residual blocks of the network, in the blue box conv2D block for convolution module, upSamping2D for sampling, the characteristics of the green box for dartnet output figure and the

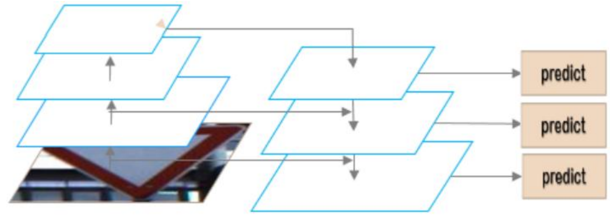

Fig -1: FPN Structure

characteristics of the sampling figure to concat feature fusion, and finally yellow box for convolution, the output of the final three characteristic figures, including the size of the convolution.

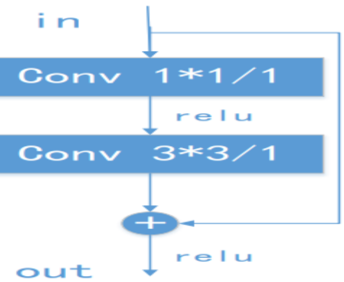

Fig -2: Residual Network

2. FPN

FPN's goal is to merge high-level and low-level feature maps in a specific way to produce a feature map with a good balance of resolution and semantic information for a better detection effect. Through lateral connection, Lin's FPN integrates high-level features with low resolution but rich semantic information with low-level characteristics with less semantic information but high resolution. FPN is mostly made up of two steps: 1. Process from the top down and side connections Bottom-up process: on the left side of the structure, the bottom-up process is prior to network transmission, forward calculation of convolution neural network channels, the process of using multiple pooling tong to extract features in order to obtain a different size chart, in the process of the propagation characteristics of figure after some layers become smaller, and some layers will not decrease the size of feature maps. Top-down process and lateral connections: as shown in the FPN pyramid structure, the sampling operation will first be on the right side of the upper figure, then left with a size 1 x 1 convolution kernels from the bottom up to generate the map feature after the convolution results with the sampling results on the fusion characteristics, the fusion specific operation FPN lateral connection structure diagram. To eliminate the aliasing effect of up-sampling, the convolution of 1*1 and 3*3 will be utilised to check each fusion result for convolution after fusion.

International Journal on Applications in Engineering and Technology
Volume 8 : Issue 2 : May 2022, pp 68 – 75 www.aetsjournal.com
ISSN (Online) : 2455 - 0523

II. The FPN YOLOV3

The FPN in YOLOV3 is not the same as the one in Aiming He's Paper. Kaiming's proposal for an FPN In his paper, he discusses the eltwise operation between high-order and low-order features after upsampling, also known as the addition fusion operation. Concat operation, i.e. the splice operation of channel direction, is a high-order feature in YOLOV3 after upsampling. FPN pyramid structure is to improve the resolution of the deep characteristic figure, enrich the semantic information of feature maps, and better forecast target, using the concat method can increase the figure characteristics of the channel number, increase the amount of calculation, and using the add method can increase the amount of information features, not increase the features of the channel number, the add method is a better choice, as shown in, the add method and concat method feat.

III. METHODOLOGY

The add fusion method and the concat fusion method have different features, and the formula below can help you comprehend the differences between the two approaches more clearly. Because each output channel's convolution kernel is generally independent, we can only view the output of a single channel. Assume the two input channels are X1, X2,..., XC and Y1, Y2,..., YC.

3.1 YOLO ALGORITHM CONCEPT

In 2015, Redmon J. first proposed the Yolo Network, which was marked by a combination of candidate box generation and classification background. While predicting, the property map is divided into 7x7 cells, and each cell is predicted, which significantly reduces the computational complexity, speeds up target detection. After a one-year break, Redmon J. again proposed YOLOv2. Compared to previous generations, the mAP of the VOC2007 test set increased from 67.4% to 78.6%. However, since a cell is responsible for predicting only one object, facing the goal of overlap, the recognition was not good enough. In comparison to RetinaNet, which has a MAP-50 of 61.1 percent, the Coco dataset's MAP-50 was enhanced from 44.0 to 57.9%. The input size for RetinaNet is 500. When the input size is 416416, the detection speed is roughly 98 ms/frame, while YOLOv3 has a detection speed of 29 ms/frame. The data is sufficient to demonstrate that YOLOv3 has achieved a very high accuracy rate while adhering to the principle of speed.

Joseph et al. suggested a new object detector, YOLOv3. Its backbone network, which includes 53 convolutional layers instead of Darknet-19, uses Darknet-53. a network's frame The hierarchical nature of the YOLOv3 network is clearly displayed.

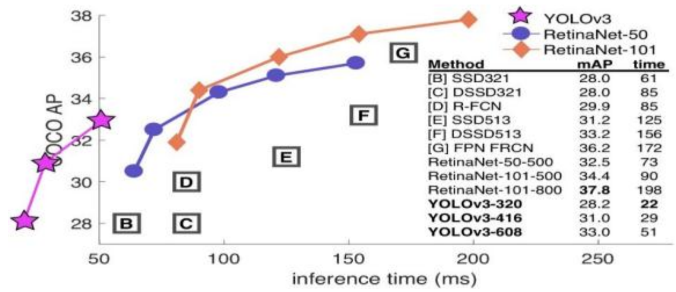

Fig -3: High accuracy rate compared to other methods

Convolution, batch normalisation, and the Leaky ReLU activation function are included in Darknet-53's smallest component DBL module. The prediction is divided into three scales: 1313, 2626, and 5252 by YOLOv3. To the detection layer, these three scales generate feature maps at three different scales. The low-level feature maps are responsible for detection and have a narrower field of vision. The deep feature map offers a vast field of vision for small targets, making it simple to locate large targets. As a result, YOLOv3 performs well in recognising both large and tiny targets. Because the YOLOv3 network has the benefits of high training efficiency, great adaptation to diverse scale targets, and suitability for complex traffic scenarios, this study improves the YOLOv3 network and applies it in this paper.

The YOLOv3 network is improved in this research, and the traffic sign data set TT100K is used for training and detection. The YOLOv3 network has the advantages of high training efficiency, great adaptation to diverse scale targets, and suitability for complex traffic scenarios.

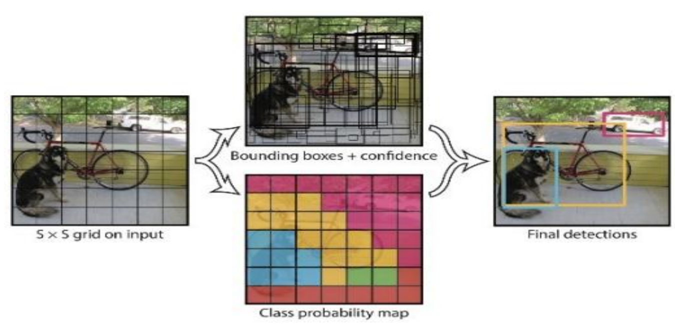

Fig -4: Output bounding box, confidence, and class probability map

International Journal on Applications in Engineering and Technology
Volume 8 : Issue 2 : May 2022, pp 68 – 75 www.aetsjournal.com
ISSN (Online) : 2455 - 0523

3.2 YOLOV3 ARCHITECTURE AND DARKNET-53

YOLO (You Only Look Once) forecasts the binding box and its possible class using a single forward pass neural network applied to the full picture. This technique speeds up the YOLO algorithm without sacrificing precision. This algorithm is available in several distinct forms on the internet. Open-source neural network framework Darknet-53 is one such example. Darknet-53 can perform the most floating-point operations per second, implying that the network structure can better utilise the GPU, making it more efficient and faster. In Figure 3, the 53 matching levels of Darknet-53 are depicted, with Convolutional and Residual layers making up the majority of the layers. By employing strides of 32, 16, and 8, we can observe that YOLO detects things in three distinct scales to accommodate varied sized items. This means that if we provide YOLOv3 a 416x416 input image, it will detect 1313, 2626, and 5252 on the scale of 1313, 2626, and 5252. YOLOv3 down samples the input image into 1313 and makes a prediction at the 82nd layer for the first scale. A 3-D tensor of dimension 1313255 is produced by the first detection scale. After that, YOLOv3 applies one convolutional layer to the feature map from layer 79 before upsampling it by a factor of two to get a size of 2626. The feature map from layer 61 is then concatenated with the upsampled feature map. The concatenated feature map is then passed through a few more convolutional layers before being exposed to the 2nd detection scale at layer 94. A 3-D tensor of dimension 2626255 is produced by the second prediction scale. The same design is used a third time to forecast the third scale. Layer 91's feature map is concatenated with a feature map from layer 36 in one convolutional layer. Layer 106 is used for the final prediction layer, resulting in a 3-D tensor with the dimension 5252255. In summary, YOLO predicts over three different scales detection, thus if we feed an image with a size of 416 416, it will provide three different output shape tensors, 1313255, 26 26 255, and 52 52 255. One extra diagram has been included to help you completely comprehend the overall architecture of the YOLOv3network. After entering the Darknet-53 network, we can see that the 416 416 input picture gets three branches.

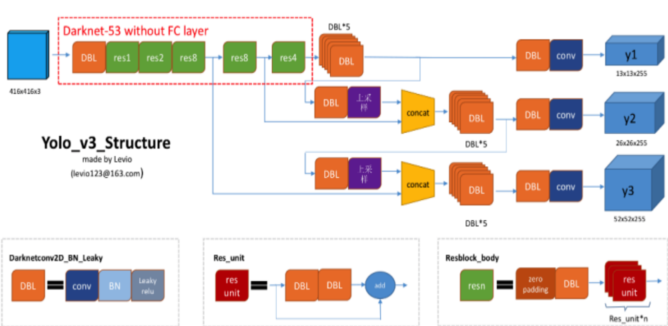

Fig -5: Overall architecture of the YOLOv3 network

3.3 YOLOV3 DETECTION

YOLO meshes input photos into thick, medium, and fine meshes to forecast large, medium, and small objects, respectively. The fat, medium, and fine grid sizes for a 416 416 input image are 1313, 2626, and 5252, respectively. It is then multiplied by 32, 16, and 8 times the length and width, accordingly.

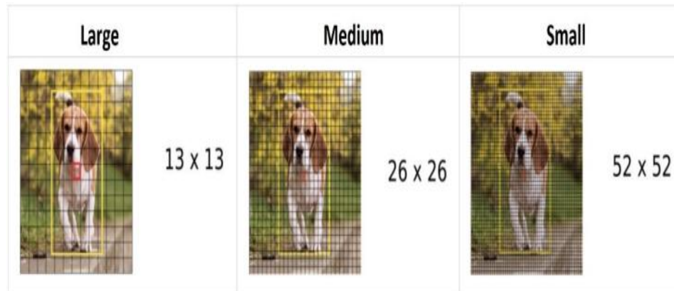

Fig -6: The grid sizes are 13×13, 26×26, and 52×52

3.4 THE LEVEL OF THE BOUNDING BOX

Three branches of the YOLOv3 network's output attributes will be given to the decode function, which will decode the map's channel information. A pre-box is represented by the black dot box, while a prediction box is represented by the blue box. The level of the box is predicted by converting output to log space and then multiplying with an anchor.

3.5 NMS PROCESSING

Finally, non-maximum suppression suppresses the most non-maximum items, as the name implies. The binding boxes with high overlap rates and low scores are removed by NMS.

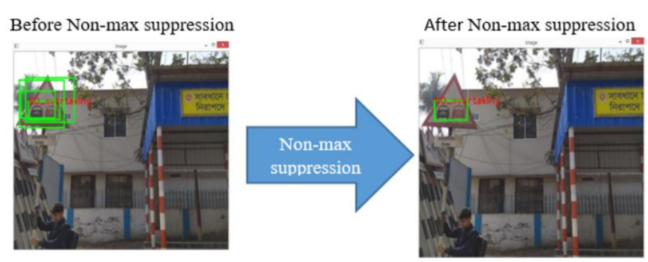

Fig -6: Non-Maximum Suppression Processing

3.6 IMPROVE YOLOV3 DETECTION ALGORITHM

The initial candidate frame and its network structure preset for the COCO data set by the YOLOv3 model are not suitable for small target detection in a real scene, and the initial candidate frame and its network structure preset for the COCO data set by the YOLOv3 model are not suitable for small target detection. To achieve traffic sign identification, this article employs K-means clustering to perform cluster analysis on the traffic sign data set, redefine the initial candidate frame size, and then improve the YOLOv3 model.

3.7 K-MEANS CLUSTER ANALYSIS

The precision and speed of target detection will be impacted if the YOLOv3 network's initial candidate block width and height are set to a fixed value. As a result, the k-means clustering technique is employed in this study to cluster the TT100K traffic sign data set, and the average degree of overlap (AvgIOU) is used as the goal cluster analysis metric. where b represents the sample, which is the ground truth target; c represents the cluster centre; nk represents the number of samples in the k cluster centre; n represents the total number of samples, and k represents the number of clusters; IOU (b, c) represents the intersection ratio between the cluster centre box and the cluster box; i represents the sample number; the sample number at the cluster centre is denoted as j. After setting k=19, the cluster analysis on the samples in the TT100K traffic sign data set is done, as well as the relationship between the k value and AvgIOU. The goal function varies steadily as the value of k increases, and the shifting inflection point can be considered the best number of initial candidate frames. Because the curve begins to become stable when the k value is bigger than 6, we chose 6 as the number of initial candidate frames.

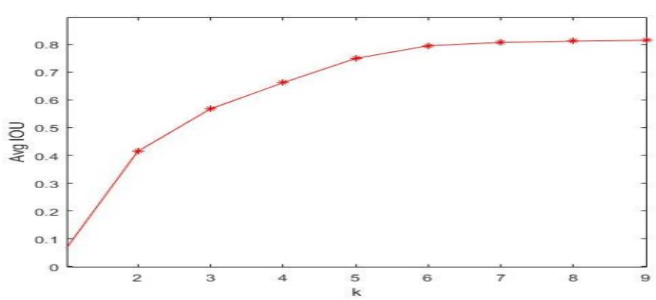

Fig -7: K-means cluster analysis results

IV. NETWORK MODEL

4.1 IMPROVE YOLOV3 NETWORK MODEL

The second down-sampling feature map in Darknet-53 is used to realise up-sampling the feature fusion map of the third scale in the YOLOv3 network, and then it is compared with Darknet the 2 times of down-sampling feature map fusion in -53 is input to the detection layer to achieve a prediction with a scale of 108108.
In the TT100K dataset, the first candidate frame corresponding to the scale of 5252 is [1314, 1920, 3032], and the initial candidate frame corresponding to the scale of 108108 is [56, 78%, 1011], the enhanced network, and the hierarchical structure is shown in figure 6. In comparison to the previous YOLOv3 algorithm, which predicts on three scales, the updated YOLOv3 algorithm only needs to forecast on two scales, allowing it to detect targets in photos more quickly. During preprocessing and multi-scale prediction, high-resolution traffic sign photos may cause loss of information or scale inconsistencies, which will damage the detection effect. Spatial pyramid pooling solves the concerns of information loss and scale inconsistencies by using separate block pooling for each picture. As a result, a fixed block size pooling operation is used before the detection layer in the spatial pyramid pooling approach. The greatest pooling core of the spatial pyramid pooling structure must be exhausted in order to achieve the fusion of the feature map level of local features and global features. It's possible that it'll be near to the size of the feature map that needs to be pooled (1313), therefore the maximum pooling core is set to 13, and the remaining two cores are reduced by 4 each, to 9 and 5. Various elements of each image are retrieved in this way to increase traffic sign detection accuracy. Despite the fact that the spatial pyramid pooling with three separate blocks increases the model's complexity and slows it down, investigations demonstrate that the model's speed decreases less and its accuracy improves more. As a result, adding the spatial pyramid pooling is desirable.

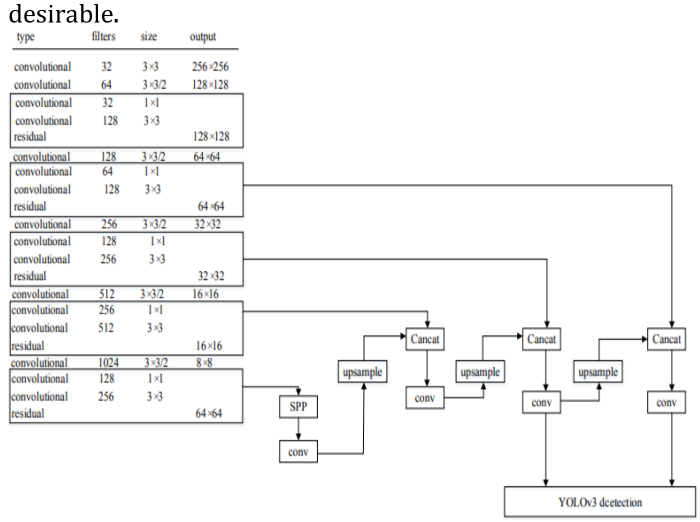

Fig -8: Improved YOLOv3 network model

4.2 YOLOv3 Coupled with Patch-Wise Detection Strategy

YOLOv3 is a convolutional neural network that, like the feature pyramid network, detects objects at three different scales of an input image. For detection, YOLOv3 employs the darknet-53 network as a feature extractor, as well as an additional seven convolutional layers at each stage. For recognising tiny items in an image, the

International Journal on Applications in Engineering and Technology
Volume 8 : Issue 2 : May 2022, pp 68 – 75 www.aetsjournal.com
ISSN (Online) : 2455 - 0523

output feature map of the deepest level is upsampled at a stride of two and concatenated with a shallower feature map. To locate different size items in a picture, this upsampling occurs twice in the network. The YOLOv3 network has batch normalisation and a Leaky ReLU activation function, which is represented by the DBL block, after each convolutional layer. When compared to other items in a photograph, traffic signs are typically smaller. The largest traffic sign is 128 by 128 pixels in the GTSDB photos of 1360 x 800 pixels. As a result, traffic signs only take up 1.5 percent of the overall pixel area in the image. The deeper networks learn about an object's nuanced look and texture, while the shallower layers learn about an object's strokes and shape aspects. Fine appearance and texture traits are less essential than form features when it comes to traffic sign detection. As a result, using an output feature map of frontal layers and avoiding deeper layers is a natural notion for detecting small items such as traffic signs. In brief, because traffic layer indications are modest, they do not require a deeper network. As a result, we advise that the network length be reduced to a tailored amount in order to identify traffic signs of various sizes with a lower miss rate. To make the network shallower, we lowered the stack of five DBL layers at each detection level to two DBL layers. The DBL block stack was reduced at each detection stage, which resulted in fewer false positives and a lower log-average miss rate (LAMR). The YOLOv3 network's mean Average Precision and overall performance will be improved from now on. As a result, we infer that five DBL layers are redundant in the network for smaller items like traffic signs, and since traffic signs are small objects in road scenes, the output feature map of deeper networks is not necessary. During the studies, we noticed that the input image size was 416 x 416 pixels for training and testing the network. The network's input image was resized to 416 x 416 pixels before being transferred to the network. It should be noted that downscaling a huge image (e.g., a 1360 x 800 pixel image in GTSDB) to 416 x 416 pixels reduces the size of little items like traffic signs.

Consider the following image, which is 1360 by 800 pixels in size and contains one traffic sign with bounding box annotations. It will have extended coordinates. Except for the last one, which surpasses 800, all of the coordinates are inside the training image's dimension constraints. As a result, it's limited to 800 characters. Figure 6a shows the patch created from the previously described image at the bottom. Similarly, the test images were sent into the network in the form of 400 by 400 pixel patches, as shown in Figure 6b. With a stride of 100 pixels, a 400 x 400 pixel window slides over the entire image. The image recorded in the window is cropped and saved in the manner displayed.

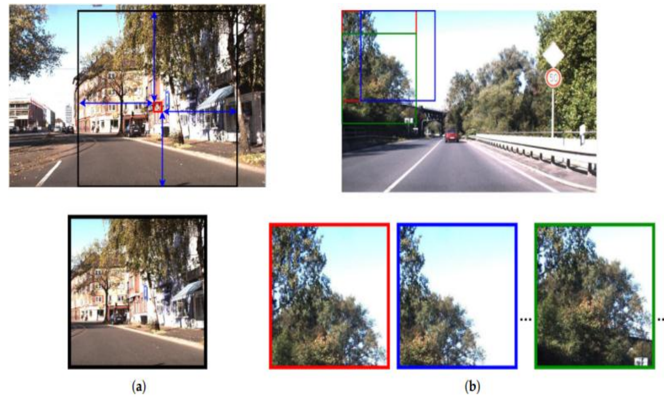

Fig -9: Patch-wise detection strategy (a) train image patch (b) test image patches.

Because of the resizing of an input image to a smaller number of pixels, the fine details of traffic signs were lost. The proposed patch-wise training assisted to keep these fine features. In comparison to the default rescaling strategy, the proposed method increased recall by 20% and subsequent detection accuracy by 13%.

4.3 Traffic Sign Recognition with You Only Look Once (Yolo) V3

This study combines Adaboost and Yolo V2 techniques for traffic sign analysis. Real traffic signs were collected in the centre of Kaohsiung, a big city in southern Taiwan, for the system. Additional research on traffic signs is given, with a focus on Taiwan. Using the proposed method for collecting the traffic signs image, this work tracks traffic signs from video recordings. The precision of the resulting dataset is checked by CNN. Focus on Taiwan's detection and identification of stop signs. They run some tests in various settings to determine the importance of anchor calculation using k-means and the original Yolo V3 for Taiwan stop sign detection and recognition. Their investigation demonstrated the importance of anchor recalculation based on our dataset. Dewi et al. look into the state-of-the-art of object detection systems like Yolo V3, Resnet 50, Densenet, and Tiny Yolo V3 when paired with spatial pyramid pooling (SPP). Their research uses the SPP principle to improve the Yolo V3, Resnet 50, Densenet, and Tiny YoloV3 backbone networks. As a result, their data demonstrate that Yolo V3 SPP has the best total BFLOPS and mAP (98.88 percent).. As a result, SPP can help all of the models in the experiment perform better. Other studies looked at other weights offered by the darknet framework, such as the best, final, and last weights. They run and analyse a comparative experiment with different weights of Yolo V3 and Yolo V3 SPP. The mean average precision (mAP) of Yolo V3 SPP is superior than other models, according to experimental results.

Based on past research, we discovered that no one had paid attention to the significance of Yolo's scale parameter in the configuration file. The importance of scale settings in the Yolo V3 and Yolo V3 SPP configuration file will be the focus of our investigation. Redmon et al. introduced Yolo V3 for the first time in 2016. The entire picture is interpreted by a single neural network. Multiscale fusion is used by Yolo V3 to produce a prediction.

Up-sample and FPN fusion are used to merge the 416 416 input image size with three scales. 13* 13, 26* 26, and 52* 52 are the three scales that were obtained. Yolo V3 was created using Darknet-53 and consists of 53 layers with deep features. Yolo V3 has outperformed ResNet-101, ResNet-152, and Darknet-19 in terms of Darknet-53 creation. The Yolo V3 algorithm divides the input image into SS grids. The grid can specify the target if the object's ground reality's central point shrinks within the appropriate grid.

4.4 YOLO V3 SPP ARCHITECTURE

This section explains how Yolo V3 with SPP can be used to recognise and identify Taiwanese road signs. The architecture of the Yolo V3 SPP. The following is the procedure for detecting objects with Yolo V3 SPP. The visual input is divided into SS grids in the first stage. According to the calculation, each grid generates K bounders. The bounding box does not contain the item if it does not have a bounding box. The object category is then chosen by the algorithm based on the category with the highest anticipated probability. Finally, Non-Maximum Suppression is used in this experiment to do maximum local exploration, suppress redundant boxes, output, and display item detection results (NMS).

Yolo V3 SPP uses convolutional layer sampling to deliver the greatest potential functionality for the max-pool layers in the study. For all photos using [route], Yolo V3 SPP uses three scales of the maximum pool. In each [route], various layers -2, -4, and -1, -3, -5, -6 in conv5 were employed. Furthermore, conv5 is the final layer of convolution, and 256 is the number of the conv5 layer filter. These feature maps, which are referred to as fixed-length representations, are then gathered. The performance of Yolo V3 and Yolo V3 SPP at different scales is compared in this experiment. The Gaussian zero-mean distributions with standard deviations of 0.01 and 0.001 are used to initialise SoftMax classification layers and boundary box regression. The parameter decay is 0.0005, the global learning rate is 0.001, and the momentum is 0.9. The learning rate parameter controls how quickly the most recent batch of data may be used to learn.

Yolo V3 1, Yolo V3 2, Yolo V3 3, Yolo V3 SPP 1, Yolo V3 SPP 2, Yolo V3 SPP 3, Yolo V3 SPP 1, Yolo V3 SPP 2, Yolo V3 SPP 3, Yolo V3 SPP 1, Yolo V3 SPP 2, Yolo V3 SPP 3, Yolo V3 SPP 1, Yolo V3 SPP For each Yolo V3 and Yolo V3 SPP, multiple scales of (0.1, 0.1), (0.2, 0.2), and (0.3, 0.3) are used, and an n-classes object detector should execute the training for a maximum of 2000n batches.

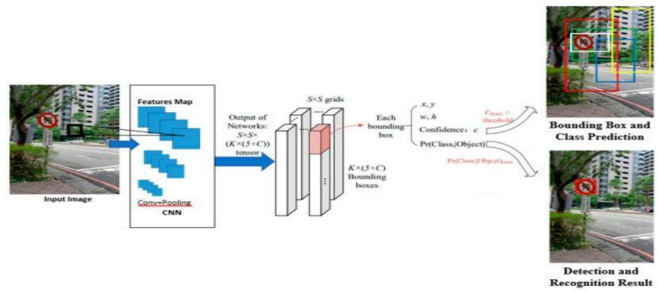

Fig -10: Yolo V3 SPP architecture

V. CONCLUSIONS

The YOLOv3-based detector and a bespoke CNN-based classification are used to develop and implement a traffic sign recognition system in this research article. On the basis of detection speed and accuracy, the detection performance we achieved is shown to be superior than earlier detector systems when considering single classes of traffic-signs. With a high mAP performance of around 45 frames/seconds and 92.2 percent accuracy, our system can reach a high detection speed. Our proposed detector can identify practically all types of traffic signs and can track the right binding box for the majority of them. To construct a state-of-the-art real-time trafficsign detection recognition system for the entire country, we want to add more trafficsign photos to our current collection from various districts in Bangladesh.

We believe that the comprehensive traffic-sign recognition pipeline we proposed will be valuable in the development of an autonomous navigation assistance for automobiles in our country in the near future. Single-stage decoders will be used in the future to identify and classify traffic marks. This strategy can assist you avoid having to use a separate network for traffic sign classification

REFERENCES

[1] Reddy, Poondru Prithvi Nath, "Driverless Car : Software Modelling and Design," September 1,2019

[2] S. P. Rajendran, L. Shine, R. Pradeep and S. Vijayaraghavan, "RealTime Traffic Sign Recognition using YOLOv3 based Detector," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-7, doi: 10.1109/ICCCNT45670.2019.8944890.

[3] Le TLe T.T., Tran S.T., Mita S., Nguyen T.D., ”Real Time Traffic Sign Detection Using Color and Shape-Based Features.,” Intelligent Information and Database Systems. ACIIDS 2010. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, vol. vol 5991., 2010.

[4] Y. Wu, Y. Liu, J. Li, H. Liu and X. Hu, ”Traffic sign detection based on convolutional neural networks,” The 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, 2013, pp. 1-7, doi: 10.1109/IJCNN.2013.6706811.

[5] Wang G.Y., Ren G.H., Jiang L.H., Quan T.F. Hole-based traffic sign detection method for traffic signs with red rim. Vis. Comput. 2014;30:539–551. doi: 10.1007/s00371-013-0879-0.

[6] Hechri, A.; Hmida, R.; Mtibaa, A. Robust road lanes and traffic signs recognition for driver assistance system. Int. J. Comput. Sci. Eng. 2015, 10, 202–209.

[7] Lillo-Castellano, J.M.; Mora-Jimenez, I.; Figuera-Pozuelo, C.; Rojo- ´ Alvarez, J.L, ”Traffic sign segmentation and classification using sta- ´ tistical learning methods.,” Neurocomputing, pp. 153, 286–299, 2015.

[8] W. Liu et al., ”SSD: Single shot multibox detector,” in Proc. Eur. Conf Comput.Vis. Cham, Switzerland: Springer, 2016.

[9] H. Guan, W. Yan, Y. Yu, L. Zhong and D. Li, ”Robust TrafficSign Detection and Classification Using Mobile LiDAR Data With Digital Images,” in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 11, no. 5, pp. 1715-1724, May 2018, doi: 10.1109/JSTARS.2018.2810143.

[10] Zhan-Li Sun, Han Wang, Wai-Shing Lau, Gerald Seet, Danwei Wang, Application of BW-ELM model on traffic sign recognition, Neurocomputing, Volume 128, 2014, Pages 153-159, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2012.11.057

[11] Kumar, A.D.; Karthika, K.; Parameswaran, L. Novel deep learning model for traffic sign detection using capsule networks. arXiv 2018, arXiv:1805.04424.

[12] Redmon, J., and A. Farhadi. ”YOLO9000: Better, faster, stronger. arXiv 2016.” arXiv preprint arXiv:1612.08242 (2016).

[13] Girshick, Ross. ”Fast r-cnn.” In Proceedings of the IEEE international conference on computer vision, pp. 1440-1448. 2015.

BIOGRAPHIES

Vasanth Kumar C is an undergraduate student, department of Computer Science and Engineering in Paavai College of Engineering.

Uma Shankar S V is an undergraduate student, department of Computer Science and Engineering in Paavai College of Engineering.